LAB # 8

USING FUNCTIONS IN C++

A function is a group of statements that together perform a task. Every C++ program has at least one function, which is **main(),** and all the most trivial programs can define additional functions. You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division usually is such that each function performs a specific task.

A function **declaration** tells the compiler about a function's name, return type, and parameters. A function **definition** provides the actual body of the function. The C++ standard library provides numerous built-in functions that your program can call. For example, function **strcat()** to concatenate two strings, function **memcpy()** to copy one memory location to another location and many more functions.

A function is known with various names like a method or a sub-routine or a procedure etc.

Defining a Function

The general form of a C++ function definition is as follows −

```
return_type function_name( parameter list ) {
   body of the function
}
```

A C++ function definition consists of a function header and a function body. Here are all the parts of a function −

Return Type − A function may return a value. The return_type is the data type of the value the function returns. Some functions perform the desired operations without returning a value. In this case, the return_type is the keyword void.

Function Name − This is the actual name of the function. The function name and the parameter list together constitute the function signature.

Parameters − A parameter is like a placeholder. When a function is invoked, you pass a value to the parameter. This value is referred to as actual parameter or argument. The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional; that is, a function may contain no parameters.

Function Body − The function body contains a collection of statements that define what the function does.

Example

Following is the source code for a function called max(). This function takes two parameters num1 and num2 and return the biggest of both −

```
// function returning the max between two numbers

int max(int num1, int num2) {
  // local variable declaration
  int result;

  if (num1 > num2)
    result = num1;
  else
    result = num2;

  return result;
}
```

Function Declarations

A function declaration tells the compiler about a function name and how to call the function. The actual body of the function can be defined separately.

A function declaration has the following parts

return_type function_name(parameter list);

For the above defined function max(), following is the function declaration −

int max(int num1, int num2);

Parameter names are not important in function declaration only their type is required, so following is also valid declaration −

int max(int, int);

Function declaration is required when you define a function in one source file and you call that function in another file. In such case, you should declare the function at the top of the file calling the function.

Calling a Function

While creating a C++ function, you give a definition of what the function has to do. To use a function, you will have to call or invoke that function.

When a program calls a function, program control is transferred to the called function. A called function performs defined task and when it's return statement is executed or when its function-ending closing brace is reached, it returns program control back to the main program.

To call a function, you simply need to pass the required parameters along with function name, and if function returns a value, then you can store returned value.

For example

```
#include <iostream>
using namespace std;

// function declaration
int max(int num1, int num2);

int main ()
{
  // local variable
  declaration: int a = 100;
  int b = 200;
  int ret;

  // calling a function to get max
  value. ret = max(a, b);
  cout << "Max value is : " << ret << endl;

  return 0;
}

// function returning the max between two
numbers int max(int num1, int num2) {
  // local variable declaration
  int result;

  if (num1 > num2)
    result = num1;
  else
    result = num2;

  return result;
}
```

```
Max value is : 200
```

TASKS TO BE PERFORMED

1. Using function, write a complete program that prints your name 10 times. The Function can take no arguments and should not return any value.

2. Write a function to calculate the factorial value of any integer entered through the keyboard.

3. Write a function which receives a float and an int from main ( ), finds the product of these two and returns the product which is printed through main ( ).

4. Given three variables x, y, z write a function to circularly shift their values to right. In other words if x = 5, y = 8, z = 10 after circular shift y = 5, z = 8, x =10 after circular shift y = 5, z = 8 and x = 10. Call the function with variables a, b, c to circularly shift values.

5. Identify the errors (if any) in your own words in the given pieces of code:

```
a)
func(int a,float b)
{
return (5.75);
}
```

```
b)
int , float newfunction(void)
{
return (7, 6.96);
}
```

6. Write a program that ask for three numbers, compare them and show the maximun. Declare a function called max_three that compares the numbers and returns the maximun.

7. Write a program that asks the user for an integer number and find the sum of all natural numbers upto that number.